IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application. No: 09/920,893	§
Filed: August 2, 2001	§
Inventor(s): David Mark Pierce and Dudley Wayne Fox, Jr.	§
Title: SYSTEM AND METHOD FOR A DELTA PAGE PROTOCOL FOR CACHING, REPLICATION, AND CLIENT/SERVER NETWORKING	§
Examiner: Bates, Kevin T.	§
Group/Art Unit: 2155	§

Atty. Dkt. No: 5150-57700

APPEAL BRIEF

Box: Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir/Madam:

Further to the Notice of Appeal filed October 26, 2007, Appellant presents this Appeal Brief. Appellant respectfully requests that this appeal be considered by the Board of Patent Appeals and Interferences.

I. REAL PARTY IN INTEREST

The subject application is owned by National Instruments Corporation, a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at 11500 N. MoPac Expressway, Bldg. B, Austin, Texas 78759-3504.

II. RELATED APPEALS AND INTERFERENCES

No related appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-3 and 5-16 are pending in the application. Claims 4 and 17-23 have been canceled. Claims 24-31 have been withdrawn.

The pending claims 1-3 and 5-16 stand rejected and are the subject of this appeal. A copy of the claims incorporating entered amendments is included in the Claims Appendix hereto.

IV. STATUS OF AMENDMENTS

All amendments have been entered. The Claims Appendix hereto reflects the current state of the claims.

V. SUMMARY OF THE INDEPENDENT CLAIMS

Independent claim 1 recites a method of logging and trending real time measurement data. The method comprises a logger application executing on a first computer system (*e.g., computer system 82 of FIG. 1*) receiving a measurement stream comprising a plurality of real time measurement data values (*e.g., block 502 of FIG. 5; p. 13, lines 15-16*), wherein the plurality of real time measurement data values are acquired from a physical system (*e.g., unit under test 150 of FIG. 2A or process 150 of FIG. 2B*) by a measurement device (*e.g., various measurement devices illustrated in FIGs. 2A and 2B*). (*Also see p. 13, lines 18-19*).

The logger application writes portions of the plurality of real time measurement data values to respective shared memory sections of a memory in the first computer system in a modular fashion (*e.g., block 504 of FIG. 5; p. 13, lines 22-24*). Each of the portions of the plurality of real time measurement data values in each of the respective shared memory sections is independently accessible (*p. 14, lines 4-6*) by a trender application executing in a second computer system (*e.g., computer system 86 of FIG. 1*).

The method comprises initiating the trender application on the second computer system (*e.g., p. 14, lines 4-6*). The trender application generates a query request for a first portion of the plurality of real time measurement data values (*e.g., block 506 of FIG. 5; p. 14, lines 10-11*).

The first computer system sends a single message to the second computer system (*block 508 of FIG. 5; p.14, lines 14-15*). The single message comprises the first portion of the plurality of real time measurement data values (*p. 14, lines 16-17*). The first portion comprises two or more of the real time measurement data values (*e.g., p. 14, lines 7-9*).

The trender application receives the single message comprising the first portion of the plurality of real time measurement data values (*p. 14, lines 18-19*) and displays the first portion of the plurality of real time measurement data values (*block 510 of FIG. 5; p. 15, lines 16-17*).

Independent claim 11 recites a method of logging and trending real time measurement data. The method comprises a logger application executing on a first computer system (*e.g., computer system 82 of FIG. 1*) writing a first plurality of real time measurement data values to a first shared memory section in the first computer system during a first time period. (*e.g., block 504 of FIG. 5; p. 13, lines 22-24*). The first plurality of real time measurement data values are acquired from a physical system (*e.g., unit under test 150 of FIG. 2A or process 150 of FIG. 2B*) by a measurement device (*e.g., various measurement devices illustrated in FIGs. 2A and 2B*). (*Also see p. 13, lines 18-19*).

The method further comprises initiating a trender application (*e.g., p. 14, lines 5-6*). The trender application executes on a second computer system (*e.g., computer system 86 of FIG. 1; p. 14, lines 5-6*). The trender application generates a query request for the first plurality of real time measurement data values and sends the query request to the first computer system (*e.g., block 506 of FIG. 5; p. 14, lines 10-13*).

The method further comprises the first computer system sending a single message to the second computer system (*block 508 of FIG. 5; p.14, lines 14-15*). The single message comprises the first plurality of real time measurement data values (*p. 14, lines 16-17*).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Section 103 Rejections

Claims 1-3 and 5-16 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Kerrigan (U.S. Patent No. 5,404,488) in view of Tacklind (U.S. Patent No. 5,626,114).

VII. ARGUMENT

Claims 1-3 and 5-16 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Kerrigan (U.S. Patent No. 5,404,488) in view of Tacklind (U.S. Patent No. 5,626,114). Appellant respectfully traverses these rejections.

Claim 1

Claim 1 recites that a logger application executes on a first computer system, and a trender application executes on a second computer system. The trender application generates a query request for a first portion of measurement data values, and the first computer system sends a single message comprising the first portion of measurement data values to the second computer system. The Examiner admits that Kerrigan does not teach that the trender application executes on a second computer system, where the logger application executes on a first computer system, as recited in claim 1. Appellant respectfully submits that the combination of Tacklind with Kerrigan also does not teach this subject matter.

Tacklind teaches that a monitor module receives measurement data values from a sensor and stores data records specifying the measurement data values (Col. 6, lines 47-60). A user can then push a button on a user interface of the monitor module to cause the monitor module to connect to a remote reporting system and transfer the data records to the remote reporting system (Col. 7, lines 20-29).

Thus, Tacklind teaches the general concept of transferring data from one computer system to another, but teaches almost nothing regarding the specific limitations recited in claim 1. In particular, Tacklind does not teach a trender application that generates a query request for a first portion of measurement data values. Instead, the monitor module simply transfers the data records to the remote reporting system in response to the user pushing a button on a user interface of the monitor module. Thus, the data records are transferred to the remote reporting system without a trender application on the remote reporting system requesting the data records or any portion of the data records. Tacklind does not teach a trender application executing on the remote reporting system or anywhere else. Since Tacklind does not even teach a trender

application that generates a query request for measurement data values, Tacklind does not, and cannot, remedy Kerrigan's deficient teaching regarding the limitation that the trender application executes on the second computer system, as recited in claim 1.

The Examiner asserts that:

> "It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Tacklind's teaching of a remote, real-time measuring station from the trender application so that measuring device can be located where the measurements must be taken, while the trender application only needs to be connected on any network."

As the Board is certainly aware, a valid 103(a) rejection requires the demonstration of a clear and particular teaching or suggestion that would motivate one of ordinary skill in the art to combine the references. However, Tacklind's disclosure is not related to the general subject matter of a logger application writing data values to memory and a trender application generating a request for the data values. Thus, Appellant respectfully disagrees that Tacklind's disclosure provides any clear teaching or suggestion for executing a logger application and a trender application such as recited in claim 1 on two different computer systems.

Thus, Appellant respectfully submits that claim 1 is patentably distinct over Kerrigan and Tacklind for at least the reasons set forth above.

Furthermore, claim 1 recites that a logger application writes portions of a plurality of real time measurement data values to respective shared memory sections in a modular fashion, where <u>each of the portions of the plurality of real time measurement data values in each of the respective shared memory sections is independently accessible by a trender application</u>. The trender application generates a query request <u>for a specific portion (i.e., the recited "first portion")</u> of the plurality of real time measurement data values and receives a single message comprising the first portion of the plurality of real time measurement data values. The Office Action relies largely upon Kerrigan to teach this subject matter.

Kerrigan relates generally to a realtime data feed engine for updating an application with the most currently received data from a data feed. The realtime engine receives data values from the data feed and caches the data values. When the application

requests updates, the realtime engine determines which of the cached data values are different from data values last sent to the application and sends only those cached data values which are determined to be different. (See Abstract, Col. 1, line 44 – Col. 2, line 39).

The Examiner has taken Kerrigan's realtime engine as the logger application recited in claim 1 and has taken the application that interfaces with the realtime engine as the trender application recited in claim 1. However, Kerrigan does <u>not</u> teach that the realtime engine writes portions of data values to respective shared memory sections <u>in a modular fashion</u>, where each of the portions of data values is <u>independently accessible</u> by the application, as recited in claim 1.

Kerrigan merely teaches that the data values are cached, but does not teach that the data values are modularized into portions or written into respective shared memory sections. With respect to the claim limitations regarding the modularized portions of measurement data values, the Examiner cites element 2016 of FIG. 7. Element 2016 (and FIG. 7 in general) relates to Kerrigan's master interest list structure. Kerrigan teaches that the realtime engine receives data from one or more data feeds, where each data feed delivers realtime data for each member of an associated group of items that are available through the data feed (Col. 1, lines 44-48). The user can configure the application to poll for data updates for only certain items associated with a given data feed (see, e.g., Col. 2, lines 40-44; Col. 5, lines 25-34; Col. 26, lines 35-38), referred to in Kerrigan as the "items of interest" or the "interest list". The master interest list structure illustrated in FIG. 7 indicates the items requested by the user (Col. 6, lines 62-66). The master interest list structure does not store the data values from the data feeds as asserted by the Examiner. Thus, Appellant disagrees that FIG. 7 (or Kerrigan in general) teaches the claim limitations regarding modularization of the measurement data values into portions that are independently accessible by a trender application.

With respect to the recited claim limitation stating that each of the portions of real time measurement data values is independently accessible by the trender application, the Examiner cites Col. 1, lines 52-55. However, Kerrigan teaches here – and throughout the disclosure – merely that the application performs a "request for updates" and in response the realtime engine determines which of the cached data values are different from the

data values that were last sent to the application and sends only those cached data values. Kerrigan does not teach writing portions of measurement data values to respective shared memory sections in a modular fashion, where each of the portions is independently accessible by a trender application.

Furthermore, Kerrigan also does not teach, "the trender application generating a query request for a first portion of the plurality of real time measurement data values". As discussed above, Kerrigan throughout the disclosure merely teaches that the application "requests updates", e.g., in order to receive new data values. Kerrigan does not teach that the application requests a specific portion (e.g., the recited "first portion") of a plurality of portions that are each independently accessible by the application. With respect to the recited limitation of the trender application generating the query request for the first portion of the plurality of real time measurement data values, the Examiner cites Col. 26, lines 32-38. However, Kerrigan simply teaches here that the application specifies the items of interest for which data updates are desired, and the application repeatedly polls the realtime engine to receive data updates for the specified items of interest. (As discussed above, the user configures the items of interest for the application.) Kerrigan does not teach that the application requests a specific portion of data values from a plurality of portions that are each independently accessible by the application, where the plurality of portions are written to respective shared memory sections in a modular fashion.

Furthermore, claim 1 also recites, "the first computer system sending a single message to the second computer system, wherein the single message comprises the first portion of the plurality of real time measurement data values, wherein the first portion comprises two or more of the real time measurement data values". With respect to these limitations of claim 1, the Examiner cites Col. 7, lines 21-30 of Tacklind. Tacklind teaches here that the data records are transferred from the monitor module to the remote reporting system, as discussed above. Here the Examiner has apparently equated the "first portion of the plurality of real time measurement data values" recited in claim 1 with the data records transferred to the remote reporting system. However, Tacklind is silent as to how the data records are transferred to the remote reporting system. In

particular, Tacklind does not teach sending the data records in a <u>single message</u>, as required by claim 1.

Thus, Appellant respectfully submits that claim 1 is patentably distinct over Kerrigan and Tacklind for at least the reasons set forth above.

<u>Claim 6</u>

Claim 6 recites the further limitations of:

> wherein the logger application receives the measurement stream and writes the portions of the plurality of real time measurement data values to respective shared memory sections of the memory at a first data rate;
>
> wherein the trender application generates the query request for the first portion of the plurality of real time measurement data values at a second data rate, wherein the second data rate is less than the first data rate.

The Examiner asserts that Kerrigan teaches these limitations at Col. 5, line 66 – Col. 6, line 3. Kerrigan teaches here that:

> API 15 and RTE 14 have distinct responsibilities in getting real-time updates into application program 18. API 15 provides mechanisms enabling the application to select the data feeds from which it desires data, to identify the items for which it wants updates and to poll the RTE for updates of the identified set of items. RTE 14 is the policeman at the intersection between feed server(s) 12 and API 15. Independently of the communication between RTE 14 and API 15, RTE 14 maintains continuous communication with feed server(s) 12 to make sure the most recent data is being obtained and updated. This occurs even if the user is not running API 15. In performing this function, RTE 14 serializes the access to each of feed server(s) 12 up to the maximum of seven. <u>To ensure that both processes can keep up with the arrival rate of the data, RTE 14 runs at time-critical priority and usually feed server(s) 12 run at the same priority</u>.

Kerrigan generally teaches that the RTE (realtime engine) receives data from the feed servers and provides data received from the feed servers to an application. (See, e.g., Col. 1, lines 44-55). As discussed above, the Examiner has equated the logger application recited in the present claims with the RTE and has equated the trender application recited in the present claims with the application that receives data updates from the RTE. The Examiner asserts that the passage quoted above teaches that, "the

first application runs at a higher priority rate thus having priority and higher speeds than the second rate)." However, the passage quoted above relates generally to the relationship between the RTE and the feed servers and teaches nothing at all about a relative rate between the RTE and the application. More particularly, Kerrigan teaches nothing at all about the RTE writing portions of data values at a first data rate and the application generating the query request for the first portion of data values at a second data rate, where the second data rate is less than the first data rate. Instead the passage relates generally to a priority at which the RTE and the feed servers (not the application) run. Furthermore, Kerrigan teaches that the RTE runs at time-critical priority, and usually the feed server(s) run at the same priority.

Thus, Appellant respectfully submits that Kerrigan and Tacklind do not teach the limitations recited in claim 6, and thus, claim 6 is separately patentable over the cited references.

Claim 7

Claim 7 recites the further limitations of:

> wherein the first computer system sending a single message to the second computer system comprises:
>
> a first observer software program executing on the first computer system querying the memory for a most recent portion of data at the second data rate; and
>
> the first observer software program sending the most recent portion of data to the second computer system at the second data rate after said querying the memory;
>
> wherein the trender application receiving the single message comprises:
>
> a second observer software program on the second computer system receiving the most recent portion of data at the second data rate from the first observer software program; and
>
> the second observer software program writing the most recent portion of data to a memory location.

Appellant notes that claim 7 depends on claim 6, which recited that the logger application (which executes on the first computer system) receives the measurement stream and writes the portions of the plurality of real time measurement data values to respective shared memory sections of the memory at a first data rate. Thus, the logger application executes on the first computer system to write the portions of data values at

the first data rate, and the first observer software program executes on the first computer system to query the memory for a most recent portion of data at the second data rate. Kerrigan simply does not teach these limitations.

With respect to the limitation of, “a first observer software program executing on the first computer system querying the memory for a most recent portion of data at the second data rate,” the Examiner again cites Col. 5, line 66 – Col. 6, line 3 (quoted above). However, as discussed above, the cited passage relates generally to a priority at which the RTE and the feed servers run. The cited passage does not teach a first observer software program executing on the first computer system querying the memory for a most recent portion of data at the second data rate.

Appellant also notes that claim 7 recites two separate steps of, “a second observer software program on the second computer system receiving the most recent portion of data” and “the second observer software program writing the most recent portion of data to a memory location”. Appellant can find no teaching in Kerrigan of the application writing data values to a memory location after receiving the data values. Kerrigan merely teaches that the RTE sends the updated data values to the application (see, e.g., Col. 33, lines 59-62), but contains no teaching as to the application writing the updated data values received from the RTE to a memory location.

Thus, Appellant respectfully submits that Kerrigan and Tacklind do not teach the limitations recited in claim 7, and thus, claim 7 is separately patentable over the cited references.

<u>Claim 8</u>

Claim 8 recites the further limitation that the memory location to which the second observer software program writes the most recent portion of data is a database. The Examiner asserts that Kerrigan teaches this limitation at Col. 6, lines 63-66. Here Kerrigan teaches that:

> Master interest list structure 200 contains all the records and fields requested by the user. It is essentially an in-memory database created from the items specified by the user in each of the applications (processes) being run. It is a union of all items requested by the user.

Thus, as discussed above with reference to claim 1, the master interest list structure stores information indicating the items requested by the user, but does not store the data values which the application receives from the RTE. Therefore, the cited portion fails to teach the limitation that the memory location to which the second observer software program writes the most recent portion of data is a database.

Thus, Appellant respectfully submits that Kerrigan and Tacklind do not teach the limitation recited in claim 8, and thus, claim 8 is separately patentable over the cited references.

Claim 9

Claim 9 recites the further limitation of, "wherein the trender application is operable to partially replicate the plurality of real time measurement data values comprising the measurement stream." In the rejection of claim 9, the Examiner asserts:

> "Column 5, lines 22 - 24, where the application takes the data values and uses a spread sheet to store and analyze the data)."

However, the cited portion of Kerrigan simply teaches that:

> Finally, the user can run up to twelve instances of an application 18. Each application 18 provides its own interface to RTE 14. In the described embodiment, the instances of the application are spreadsheet programs.

Appellant can find no teaching here, or elsewhere in Kerrigan, of the application partially replicating the plurality of real time measurement data values, as recited in claim 9.

Thus, Appellant respectfully submits that Kerrigan and Tacklind do not teach the limitation recited in claim 9, and thus, claim 9 is separately patentable over the cited references.

Claim 10

Claim 10 recites the further limitations of:

> wherein the logger application writing portions of the plurality of real time measurement data values to respective shared memory sections of a memory in the first computer system in a modular fashion comprises:

> creating a header record comprising a series of bits, wherein the bits in the header record indicate a changed status of the respective shared memory sections;
>
> the logger application writing the header record in the shared memory.

The Examiner asserts that Kerrigan teaches these limitations at Col. 6, lines 9-23. However, the cited portion of Kerrigan recites as follows:

> If the data element does exist in the in-memory database, then that data element is updated.
>
> RTE 14 maintains counters and flags to provide four levels of checking to determine if any given data value has changed since the last update of that instance of an application 18. Only the most recent, changed values are written into an instance of application 18 when the application requests updates. The four levels of checking are:
> 1. Has a value for a given data feed been updated?
> 2. Has some field within a given key been updated?
> 3. Has a given field value been updated?
> 4. Has the data value for the field actually changed from the previous value that was written to the worksheet?

Thus, the cited portion relates generally to determining if any given data value has changed since the last update. However, Kerrigan teaches nothing whatsoever regarding the specific limitations recited in claim 10 of creating a header record comprising a series of bits, where the bits in the header record indicate a changed status of the respective shared memory sections, and the logger application writing the header record in the shared memory.

Thus, Appellant respectfully submits that Kerrigan and Tacklind do not teach the limitations recited in claim 10, and thus, claim 10 is separately patentable over the cited references.

<u>Claim 11</u>

Independent claim 11 recites similar limitations as discussed above with reference to claim 1. Appellant respectfully submits that claim 11 is patentably distinct over Kerrigan and Tacklind for reasons similar to those discussed above.

For example, claim 11 recites in pertinent part, "a logger application executing on a first computer system…" and "the trender application executing on a second computer

system…" The cited references, taken either singly or in combination, do not teach executing a logger application and a trender application on two different computer systems, for reasons similar to those set forth above with respect to claim 1.

Claim 11 also recites the limitation of, "the first computer system sending a <u>single message</u> to the second computer system, wherein the <u>single message</u> comprises the first plurality of real time measurement data values." The cited references, taken either singly or in combination, also fail to teach this further limitation, for reasons similar to those set forth above with respect to claim 1.

Appellant thus respectfully submits that claim 11 is also patentably distinct over Kerrigan and Tacklind.

<u>Claim 12</u>

Claim 12 recites the further limitations of:

> performing a single write operation in the second computer system to store the first plurality of real time measurement data values in a memory of the second computer system.

In the rejection of claim 12, the Examiner asserts:

> "Column 5, lines 22 - 24, where the application takes the data values and uses a spread sheet to store and analyze the data)."

However, the cited portion of Kerrigan simply teaches that:

> Finally, the user can run up to twelve instances of an application 18. Each application 18 provides its own interface to RTE 14. In the described embodiment, the instances of the application are spreadsheet programs.

This teaches nothing whatsoever regarding the specific limitations recited in claim 12 of performing a <u>single write operation</u> in the second computer system to store the first plurality of real time measurement data values in a memory of the second computer system.

Thus, Appellant respectfully submits that Kerrigan and Tacklind do not teach the limitations recited in claim 12, and thus, claim 12 is separately patentable over the cited references.

Claim 13

Claim 13 recites the further limitations of, "updating a local cache in a memory of the second computer system with the first plurality of real time measurement data values using a single write operation." In the rejection of claim 13, the Examiner cites Col. 1, lines 63-66. Kerrigan teaches here that:

> In general, in one aspect, the invention is a realtime engine for interfacing one or more data feeds with an application where each of the one or more data feeds delivers realtime data for each member of an associated group of items that are available through the data feed. The interface includes means for caching the most recent data values received from a selected one of the one or more data feeds for at least some members of the associated group of items for the selected data feed; and means for sending at least some of the cached data values to the application in response to a request for updates.
>
> Preferred embodiments include the following features. The realtime engine includes means for identifying a subset of the associated group of items for the selected data feed for which the application desires data. The subset identifies the members of the associated group for which the caching means caches the most recent data values. Also, the realtime engine includes means for determining which of the cached data values are different from data values for corresponding items last sent to the application. The sending means sends only those cached data values which are determined to be different. In addition, the realtime engine includes means for storing copies of the data values last sent to the application. The determining means compares contents of the storing means with contents of the caching means to determine which data values are different from those last sent to the application.

Thus, the cited portion of Kerrigan relates to data values cached by the realtime engine. However, Kerrigan does not teach updating a cache with the first plurality of real time measurement data values using a single write operation, as recited in claim 13.

Moreover, claim 13 recites updating a local cache in a memory of the second computer system with the first plurality of real time measurement data values. However, in the Kerrigan/Tacklind combination proposed by the Examiner, the data values would be cached by the realtime engine on the first computer system, not the second computer system. Kerrigan and Tacklind, taken either singly or in combination, do not teach updating a local cache in a memory of the second computer system with the first plurality of real time measurement data values using a single write operation, as recited in claim 13.

Thus, Appellant respectfully submits that Kerrigan and Tacklind do not teach the limitations recited in claim 13, and thus, claim 13 is separately patentable over the cited references.

Claim 15

Claim 15 recites the further limitations of, "wherein the first computer system sending a single message to the second computer system comprises the first computer system sending a single network message to the second computer system." In regard to sending a single network message, the Examiner cites Col, 7, lines 21-30. Tacklind teaches here that the data records are transferred from the monitor module to the remote reporting system. However, Tacklind does not teach that the data records are transferred in a single network message. Appellant respectfully submits that it is most likely that the data records would be transferred using multiple network messages.

Thus, Appellant respectfully submits that Kerrigan and Tacklind do not teach the limitations recited in claim 15, and thus, claim 15 is separately patentable over the cited references.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-3 and 5-16 was erroneous, and reversal of the decision is respectfully requested.

The fee of $510.00 for filing this Appeal Brief is being paid concurrently via EFS-Web. If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above-referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. The Commissioner is hereby authorized to charge any fees which may be required or credit any overpayment to Meyertons, Hood, Kivlin, Kowert & Goetzel P.C., Deposit Account No. 50-1505/5150-57700/JCH.

Respectfully submitted,

/Jeffrey C. Hood/
Jeffrey C. Hood, Reg. #35198
ATTORNEY FOR APPLICANT(S)

Meyertons Hood Kivlin Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: 2007-12-21 JCH/JLB

IX. CLAIMS APPENDIX

The following lists the claims as incorporating entered amendments, and as on appeal.

1. (Previously Presented) A method of logging and trending real time measurement data, the method comprising:

a logger application executing on a first computer system receiving a measurement stream comprising a plurality of real time measurement data values, wherein the plurality of real time measurement data values are acquired from a physical system by a measurement device;

the logger application writing portions of the plurality of real time measurement data values to respective shared memory sections of a memory in the first computer system in a modular fashion;

wherein each of the portions of the plurality of real time measurement data values in each of the respective shared memory sections is independently accessible by a trender application executing in a second computer system;

initiating the trender application on the second computer system;

the trender application generating a query request for a first portion of the plurality of real time measurement data values;

the first computer system sending a single message to the second computer system, wherein the single message comprises the first portion of the plurality of real time measurement data values, wherein the first portion comprises two or more of the real time measurement data values;

the trender application receiving the single message comprising the first portion of the plurality of real time measurement data values;

the trender application displaying the first portion of the plurality of real time measurement data values.

2. (Previously Presented) The method of claim 1, wherein each of the portions of the plurality of real time measurement data values in each of the respective shared

memory sections is independently accessible by a trender application executing in a second computer system using a single network message.

3. (Previously Presented) The method of claim 1, wherein each of the portions of the plurality of real time measurement data values in each of the respective shared memory sections independently and accurately represents a subset of the measurement stream.

4. (Cancelled)

5. (Original) The method of claim 4, wherein the single message is a delta page.

6. (Previously Presented) The method of claim 4,

wherein the logger application receives the measurement stream and writes the portions of the plurality of real time measurement data values to respective shared memory sections of the memory at a first data rate;

wherein the trender application generates the query request for the first portion of the plurality of real time measurement data values at a second data rate, wherein the second data rate is less than the first data rate.

7. (Original) The method of claim 6,

wherein the first computer system sending a single message to the second computer system comprises:

a first observer software program executing on the first computer system querying the memory for a most recent portion of data at the second data rate; and

the first observer software program sending the most recent portion of data to the second computer system at the second data rate after said querying the memory;

wherein the trender application receiving the single message comprises:

a second observer software program on the second computer system receiving the most recent portion of data at the second data rate from the first observer software program; and

the second observer software program writing the most recent portion of data to a memory location.

8. (Original) The method of claim 7, wherein the memory location is a database.

9. (Previously Presented) The method of claim 1, wherein the trender application is operable to partially replicate the plurality of real time measurement data values comprising the measurement stream.

10. (Previously Presented) The method of claim 1, wherein the logger application writing portions of the plurality of real time measurement data values to respective shared memory sections of a memory in the first computer system in a modular fashion comprises:

creating a header record comprising a series of bits, wherein the bits in the header record indicate a changed status of the respective shared memory sections;

the logger application writing the header record in the shared memory.

11. (Previously Presented) A method of logging and trending real time measurement data, the method comprising:

a logger application executing on a first computer system writing a first plurality of real time measurement data values to a first shared memory section in the first computer system during a first time period, wherein the first plurality of real time measurement data values are acquired from a physical system by a measurement device;

initiating a trender application;

the trender application executing on a second computer system generating a query request for the first plurality of real time measurement data values and sending the query request to the first computer system;

the first computer system sending a single message to the second computer system, wherein the single message comprises the first plurality of real time measurement data values.

12. (Previously Presented) The method of claim 11, further comprising:

performing a single write operation in the second computer system to store the first plurality of real time measurement data values in a memory of the second computer system.

13. (Previously Presented) The method of claim 12, wherein said performing a single write operation comprises:

updating a local cache in a memory of the second computer system with the first plurality of real time measurement data values using a single write operation.

14. (Previously Presented) The method of claim 12, further comprising:

the trender application reading the first plurality of real time measurement data values from the memory of the second computer system after said performing a single write operation in the second computer system to store the first plurality of real time measurement data values in a memory of the second computer system.

15. (Original) The method of claim 12, wherein the first computer system sending a single message to the second computer system comprises the first computer system sending a single network message to the second computer system.

16. (Previously Presented) The method of claim 12, further comprising:

the logger application executing on the first computer system writing second and subsequent pluralities of real time measurement data values to second and subsequent shared memory sections in the first computer system during second and subsequent time periods;

wherein the first, second and subsequent pluralities of real time measurement data values affect only what is written to their respective first, second and subsequent shared memory sections.

17-23. (Cancelled)

24. (Withdrawn) A method of processing measurement data, the method comprising:

(a) receiving first measurement data of a first data type of a plurality of data types from a first measurement device of a plurality of measurement devices;

(b) storing the received measurement data in a shared memory location;

(c) appending one or more bits to a first header record wherein the header record comprises a series of bits, wherein each bit in the series of bits represents a section of the stored measurement data in the shared memory location;

(d) retrieving at least a subset of the stored measurement data from the shared memory location substantially concurrently with the measurement data being stored in the shared memory location;

repeating (a) - (d) for second and subsequent measurement data wherein a second header record is created when the first header record reaches a user specified number of bits and subsequent header records are created when the second header record reaches the user specified number of bits.

25. (Withdrawn) The method of claim 24, wherein measurement data is received for a user specified time interval.

26. (Withdrawn) The method of claim 24,

wherein each bit in each header record is designated as "changed" for those sections of stored measurement data which comprise any change in the measurement data from previously stored measurement data from a same measurement device;

wherein each bit in each header record is designated as "not changed" for those sections of stored measurement data which comprise no change in the measurement data from previously stored measurement data from a same measurement device.

27. (Withdrawn) The method of claim 24, wherein in retrieving at least a subset of the stored measurement data from the shared memory location substantially concurrently with the measurement data being stored in the shared memory location, measurement data associated with a single header record is retrieved.

28. (Withdrawn) The method of claim 24, wherein in retrieving at least a subset of the stored measurement data from the shared memory location substantially concurrently with the measurement data being stored in the shared memory location, measurement data associated with one or more header records is retrieved.

29. (Withdrawn) The method of claim 24, wherein the first measurement data comprises live data acquired from a data acquisition device.

30. (Withdrawn) The method of claim 24, wherein the first measurement data comprises one or more of: waveform data; single-point data, wherein single-point data comprises a data value and a data timestamp; alarm data; event data.

31. (Withdrawn) The method of claim 24, wherein the first measurement data comprises measurement data acquired from a measurement device.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.